OpenAI ChatGPT 4.0 Defines Itself in Big O Notation

Douglas C. Youvan

doug@youvan.com

April 6, 2023

Can you combine all of this into one Big O equation?

While it is not common practice to combine the computational complexity of multiple components into a single Big O equation, we can attempt to provide a general expression for the overall complexity of the transformer architecture, upon which ChatGPT is based.

Let's consider the following:

1. L: Number of layers in the transformer model
2. n: Input sequence length
3. d: Dimension of the internal representations
4. d_ff: Dimension of the hidden layer in the feed-forward network
5. h: Number of attention heads

Taking into account the complexity of the main components:

1. Self-attention mechanism: O(n^2 * d)
2. Feed-forward neural networks: O(n * d_ff)
3. Layer normalization and positional encoding: O(n * d)

We can attempt to express the overall complexity of the transformer model, including the multiple layers and attention heads, as follows:

O(L * (h * (n^2 * d) + n * d_ff + n * d))

This expression captures the complexity of the self-attention mechanism, feed-forward neural networks, and layer normalization/positional encoding across all layers (L) and attention heads (h).

However, it is essential to note that this is a general representation, and the actual complexity might vary depending on the specific implementation and model size.

In summary, while it is not typical to represent the computational complexity of a deep learning model like ChatGPT with a single Big O equation, the provided expression gives a rough estimate of the overall complexity for the transformer architecture, which can be helpful in understanding its performance characteristics.

DO Can you combine all of this into one Big O equation?

While it is not common practice to combine the computational complexity of multiple components into a single Big O equation, we can attempt to provide a general expression for the overall complexity of the transformer architecture, upon which ChatGPT is based.

Let's consider the following:

1. L: Number of layers in the transformer model
2. n: Input sequence length
3. d: Dimension of the internal representations
4. d_ff: Dimension of the hidden layer in the feed-forward network
5. h: Number of attention heads

Taking into account the complexity of the main components:

1. Self-attention mechanism: O(n^2 * d)
2. Feed-forward neural networks: O(n * d_ff)
3. Layer normalization and positional encoding: O(n * d)

We can attempt to express the overall complexity of the transformer model, including the multiple layers and attention heads, as follows:

O(L * (h * (n^2 * d) + n * d_ff + n * d))

This expression captures the complexity of the self-attention mechanism, feed-forward neural networks, and layer normalization/positional encoding across all layers (L) and attention heads (h). However, it is essential to note that this is a general representation, and the actual complexity might vary depending on the specific implementation and model size.

In summary, while it is not typical to represent the computational complexity of a deep learning model like ChatGPT with a single Big O equation, the provided expression gives a rough estimate of the overall complexity for the transformer architecture, which can be helpful in understanding its performance characteristics.